

Flare-1: A Merged Hybrid Architecture for Grounded, Tool-Using Small Language Models

North ML

<https://huggingface.co/North-ML1>

October 2026

Abstract

We describe Flare-1, the architecture behind North ML’s Ink-1 (~725M parameters), Fiber-1, and Fiber-1 Pro (~1.63B parameters). Flare-1 couples a Qwen3.5-derived gated DeltaNet/GQA hybrid backbone (“Starlight”) with a second, smaller transformer trunk (“Fiber”) that shares the backbone’s embedding table and reads its hidden states through a learned bridge. Two additions give the merged system its character: **AgentRegisters**, small persistent memory slots attached to the backbone’s full-attention layers whose state survives across inference calls, and **DeltaMemory**, delta-rule associative memory layers inside the Fiber trunk. A linear router head on the pooled embedding gates a retrieval-and-tool pipeline, and a state-management layer (“Era”) provides lossless snapshots, rollback, and prefix checkpoints for speculative decoding and long-context work. All components ship in a single `model.safetensors` artifact, 8-bit quantized, loadable on commodity hardware (a 16 GB Mac Mini). We report measured benchmark results and a 31-case held-out reliability suite, and we include a controlled 1M-parameter comparison against a parameter-matched GPT baseline whose result is negative: the mechanism set shows no significant advantage at that scale.

1 Introduction

Small open models are fluent but unreliable: stale knowledge, hallucinated facts, no tools. The premise of Flare-1 is that much of this unreliability is *systemic*, not parametric—a small model wrapped in correct routing, retrieval, and verification can answer grounded questions far above its raw weight class, while remaining cheap enough to run on a laptop.

Flare-1 is the merger of two prior North ML lines:

- **Starlight** — a Qwen3.5-style hybrid backbone (gated DeltaNet linear attention + periodic full GQA attention), distilled and light-pruned from a larger teacher.
- **Fiber** — a 158M-parameter trunk originally built as a router/draft model, extended with delta-rule associative memory.

The architecture goal: keep the backbone’s trained behavior intact (all additions are zero-init or side-path) while adding persistence (registers), associative recall (DeltaMemory), and decision-making (router) as separable, measurable components.

	Ink-1	Fiber-1 / Pro
Layers	24	24
Hidden size	1024	2048
Pattern	3× linear_attention + 1× full_attention (interval 4)	
Attention	GQA, 8 query / 2 KV heads, head_dim 256	
MLP intermediate	3584	scaled SwiGLU
Vocabulary	65,536	65,536
Parameters	565,222,208	1,413,111,616

Table 1: Backbone configuration shared across tiers.

2 Architecture

2.1 Starlight backbone

Linear-attention layers are gated DeltaNet blocks. Each maintains a per-head recurrent state $S_t \in \mathbb{R}^{d \times d}$ updated by the delta rule with a learned per-step gate:

$$S_t = S_{t-1} \left(I - \beta_t k_t k_t^\top \right) + \beta_t k_t v_t^\top, \quad (1)$$

where k_t, v_t are projected from the input and $\beta_t \in (0, 1)$ is the learned gate controlling erasure strength. Every fourth layer is full GQA attention with RoPE, which keeps per-token cost close to linear attention while retaining periodic exact retrieval over the sequence.

2.2 AgentRegisters

After each full-attention layer, a small set of register slots is updated recurrently:

$$m_t = \tanh(W_r[h_t; m_{t-1}]), \quad (2)$$

$$\tilde{h}_t = h_t + W_o m_t, \quad W_o^{(0)} = 0. \quad (3)$$

Because W_o is zero-initialized, registers are functionally invisible to the trained backbone at initialization and only contribute once trained. Crucially, register state is *persisted across inference calls*, not just across tokens within a sequence: it acts as a session-level recurrent memory that survives generation boundaries, snapshots, and rollbacks. Register state is a first-class citizen of the runtime state—it is saved, restored, and prefix-cached alongside KV and DeltaNet states.

Parameter cost is small: 295,008 on Ink-1 and 589,920 on Fiber-1.

2.3 Fiber trunk and DeltaMemory

The Fiber trunk is a complete 24-layer transformer (hidden 768, 12 heads, intermediate 2048; 158,143,524 parameters) that shares the backbone’s 65,536-token embedding table. Three of its layers are replaced by **DeltaMemory** layers: delta-rule associative memory maintaining an $S \in \mathbb{R}^{H \times D \times D}$ matrix state read and written each step—the same update family as DeltaNet, but dedicated to key→value association rather than sequence modeling.

Fiber serves three roles:

1. **Router** — pooled output feeds the search/think head (§2.4).

2. **Draft model** — cheap candidate tokens for speculative decoding against Starlight verification.
3. **Associative memory** — DeltaMemory state persists across calls like the registers.

2.4 Bridge and router head

A learned linear bridge (1,572,864 parameters on Ink-1; 3,145,728 on Fiber-1) projects Starlight hidden states into Fiber’s width. The router head is a 1,538-parameter linear probe over the pooled representation emitting p_{search} and p_{think} ; a threshold on p_{search} gates the retrieval pipeline. The router is deliberately simple—contextual rewriting upstream turns follow-ups into standalone questions before routing, and raw-message overrides keep small talk from ever triggering search.

2.5 Era: state management, not attention

Era is a lossless state manager rather than a new attention mechanism. It snapshots and restores the complete runtime state—KV cache, DeltaNet recurrent states, AgentRegister memory, DeltaMemory matrices, and Fiber state—enabling:

- **Rollback**: rewind generation to any prior checkpoint.
- **Prefix checkpoints**: resume from a shared prompt prefix without re-prefill (disabled by default pending parity verification).
- **Speculative decoding**: Fiber drafts, Starlight verifies, mismatches roll back exactly.

2.6 Merged artifact

Each tier is one `model.safetensors` file (~ 755 MB / ~ 1.76 GB) containing all 1,368 tensors plus JSON metadata: backbone and trunk configs, router thresholds, search config, generation defaults, prompt templates, chat template, and tokenizer. Weight matrices use block-quantized 8-bit storage ([64, 8] scheme, packed as U32); support tensors remain BF16/F32. A GGUF container build is published for tooling that prefers the format (identical tensors; not llama.cpp-compatible).

3 Training

- **Starlight (Fiber-1/Pro)**: the ~ 1.63 B backbone is produced by 95.8% light pruning plus distillation from a larger Qwen3.5-style teacher.
- **Ink-1**: a 565M Starlight-small variant trained with the same recipe.
- **Fiber-1 Pro**: Fiber-1 weights plus online GRPO reinforcement learning and a neural fact-verification pass that checks emitted claims against retrieved sources and revises unsupported ones.
- The Fiber trunk and router are trained as a draft-and-route auxiliary; the bridge is learned on embedding-paired data.

4 Retrieval and agentic pipeline

At inference, the Kestrel runtime executes: last-user-message → contextual rewrite (history-aware standalone question) → raw-message chat overrides → router → tools (weather, geocoding, time, math, Python execution) → retrieval (model-written query → Wikipedia + live web → BM25 + embedding-similarity rerank with recency and disambiguation weighting) → grounded generation with source citations → optional verification pass (Pro) → streamed pipeline stages to the client.

5 Evaluation

5.1 Standard benchmarks (measured)

Benchmark	Ink-1 (725M)	Fiber-1 (1.63B)
ARC-Challenge (1165q)	63.6%	73.5%
MMLU (280q subset)	39.3%	48.9%
HumanEval (164)	18.3%	37.2%
GSM8K (300q)	40.3%	unstable [†]

Table 2: Measured benchmark results. [†]GSM8K varied sharply across runs; treat as noise.

5.2 Held-out reliability suite

31 cases spanning chat, identity, grounding, follow-up, pushback, reasoning, and code, using fictional entities and offline retrieval fixtures, judged semantically:

	Baseline runtime	Candidate runtime
Ink-1	15/20	17/20
Fiber-1	14/20	17/20
Fiber-1 Pro	15/20	19/20

Table 3: Held-out reliability scores (test split of 20 cases). The candidate runtime fixed follow-up handling (1/3 → 3/3) and Pro pushback (0/2 → 2/2) with no regressions.

5.3 Flare-1M controlled experiment (negative result)

A 965,908-parameter Flare-mini (DeltaNet + GQA + registers + DeltaMemory + cross-chunk state) versus a 981,312-parameter GPT baseline on identical data and steps:

The architecture additions are not yet proven to help at small scale—a longer-context training run is needed to discriminate.

6 Limitations

- English only; hallucination remains possible, especially under the small Ink-1 writer, which can blend corrected and stale claims.
- The router is a linear probe—coverage errors route wrong.

Metric	Flare-1M	GPT baseline
Held-out CE	0.0004	0.0003
Passkey @ depth .1/.5/.9	0 / .083 / .083	0 / 0 / .083
Associative recall	.146	.104
Train time	576 s	36 s

Table 4: 1M-parameter controlled comparison. No significant advantage; the passkey task (2,600 tokens) exceeded the 384-token training window for both models, and near-zero held-out CE indicates corpus duplication.

- GSM8K scores are unstable across runs.
- Era prefix caching is disabled by default pending parity tests.
- No evidence the merged system beats frontier models on general capability; the claim is capability per dollar and grounded verifiability.

7 Related work

Gated linear attention (DeltaNet and the delta-rule family; the Qwen3.5 hybrid backbone); recurrent + full-attention hybrids (Jamba, Zamba); retrieval-augmented generation; speculative decoding; register/memory tokens in transformer architectures.

8 Availability

Weights: North-ML1/Ink-1, North-ML1/Fiber, North-ML1/Fiber-pro (safetensors); North-ML1/{Ink-1,Fiber-1} (container builds); North-ML1/Flare-Preview (architecture card and POC code/results). Runtime: Kestrel (MLX). All artifacts Apache-2.0.